FINANCIAL FRAUD DETECTION MACHINE LEARNING MODEL

This project aimed to develop a machine learning model for fraud detection in financial transactions. A Random Forest model was implemented to analyse transactional data and identify fraudulent patterns.

Project Report

Table of Contents

1. Introduction

Financial fraud poses a significant threat to individuals, businesses, and the global economy. It encompasses a wide range of illegal activities, including credit card fraud, identity theft, money laundering, and insider trading. These fraudulent actions can result in substantial financial losses, damage to reputations, and erosion of trust in financial institutions.

Accurate fraud detection is crucial for mitigating these risks. By identifying fraudulent activities early on, organizations can prevent financial losses, protect their customers, and maintain their integrity. Effective fraud detection systems can help to deter fraudulent behaviour and reduce the overall prevalence of financial crime.

1.1 Goal

The primary goal of this project is to develop a robust machine learning model that can accurately and proactively identify fraudulent transactions within a financial dataset. The model should be able to analyse various transaction attributes and patterns to detect anomalies indicative of fraudulent behaviour, thereby enabling timely intervention and prevention of financial losses.

1.2 Setup

Features and Their Descriptions:

- **step:** Represents the time elapsed since the simulation began, measured in hours.
- **type:** Indicates the transaction type, such as CASH-IN, CASH-OUT, DEBIT, PAYMENT, or TRANSFER.
- **amount:** Represents the transaction amount in the local currency.
- **nameOrig:** Identifies the customer who initiated the transaction.
- **oldbalanceOrg:** Shows the initial balance of the originating customer before the transaction.
- **newbalanceOrg:** Displays the new balance of the originating customer after the transaction.
- **nameDest:** Identifies the recipient of the transaction.
- **oldbalanceDest:** Indicates the initial balance of the recipient before the transaction (not available for merchants).
- **newbalanceDest:** Shows the new balance of the recipient after the transaction (not available for merchants).
- **isFlaggedFraud:** A binary flag indicating whether the transaction was flagged as suspicious due to a large transfer amount (over 200,000).

Target variables and its description:

- **isFraud:** A binary flag indicating whether the transaction is fraudulent (1) or legitimate (0).

Dataset Summary:

- Total Transactions: **63,626,200**
- Target Variable: **isFraud**
- Features: step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrg, nameDest, oldbalanceDest, newbalanceDest, isFlaggedFraud

2. Data Exploration and Preprocessing

2.1 Data Loading and Understanding

Loading the Dataset: The dataset was loaded into a pandas DataFrame using the **pd.read_csv** function

Data Inspection: Summarize the initial inspection of the data using the results of **df.info()**

Data Summary: Summary statistics for numerical features were obtained using the **describe()** method. These provide insights into the distribution of values, such as mean, median, standard deviation, minimum, maximum and percentiles

2.2 Handling Missing Values

To assess the completeness of the dataset and identify any missing values, the **isnull()** function in Pandas was utilized. This function returns a Boolean mask indicating which elements in the DataFrame are missing.

2.3 Correlation Analysis

Correlation analysis is a statistical technique that measures the degree to which two variables change together. It can be useful in identifying redundant or highly correlated features in a dataset. In this dataset, correlation between the various features were determined and two pairs of features are strongly correlated (correlation coefficient greater than or equal to 0.8).

These features are:
1) ‘**newbalanceDest**’ and ‘**oldbalanceDest**’

2) ‘**oldbalanceOrg**’ and ‘**newbalanceOrig**’

2.4 Feature Engineering

To ensure compatibility with machine learning algorithms, certain features were transformed:

- **One-Hot Encoding:** The “**type**” column was one-hot encoded to convert categorical variables into numerical representations.
- **Dropping Irrelevant Columns:** The **nameOrig** and **nameDest** columns were dropped as they were deemed irrelevant for fraud detection in this specific context.

3. Model Selection and Training

Random Forest is a versatile and powerful machine learning algorithm used primarily for classification and regression tasks. It works by creating multiple decision trees during training and combining their outputs to improve accuracy and prevent overfitting.

Here’s a brief overview:

- **Decision Trees**: At its core, a random forest builds multiple decision trees, which are simple models that split data into branches based on certain features to make predictions.
- **Ensemble Learning**: Random Forest is an ensemble method, meaning it combines multiple models to produce better results than any single model could.
- **Randomness**: Each tree in the forest is built from a random subset of the data and features, ensuring that the trees are diverse.
- **Voting**: For classification tasks, the forest predicts the class by taking a majority vote from all the trees. For regression, it averages the predictions.

Advantages include high accuracy, robustness to noise, and reduced overfitting, making Random Forest a go-to method for many practical problems.

Why Random Forest is a Suitable Choice for Fraud Detection?

Random Forest is a powerful machine learning algorithm that is well-suited for fraud detection tasks, especially when dealing with complex datasets and imbalanced classes. Here's why it's a good choice for this specific dataset:

1. Handling Multiple Features:

- This dataset includes a variety of features (e.g., transaction type, amount, time), making it a good candidate for a model that can handle multiple inputs effectively. Random Forest is excellent at capturing complex relationships between features.

2. Non-Linear Relationships:

- Fraudulent activities often involve non-linear patterns. Random Forest can model non-linear relationships between features, making it suitable for detecting complex fraud patterns.

3. Robustness to Noise and Outliers:

- Real-world fraud data can be noisy and contain outliers. Random Forest is less sensitive to noise and outliers compared to some other algorithms, making it more robust.

4. Feature Importance:

- Random Forest provides feature importance scores, which can help one identify the most significant factors contributing to fraudulent transactions. This can be valuable for understanding the underlying mechanisms of fraud and improving one's fraud detection strategy.

5. Handling Imbalanced Classes:

- Fraudulent transactions are often rare compared to legitimate ones, leading to imbalanced datasets. Random Forest can handle imbalanced classes effectively, especially when combined with techniques like class weighting or oversampling.

6. Interpretability:

- While not as interpretable as some other algorithms, Random Forest can still provide insights into the decision-making process through feature importance scores and partial dependence plots. This can be helpful for understanding the model's behaviour and identifying potential biases.

3.1 Baseline Models:

This section presents the baseline models developed for the fraud detection task. The models are based on Random Forest, a popular machine learning algorithm known for its ability to handle complex datasets and non-linear relationships. To establish a benchmark for performance, four baseline models were created by removing different combinations of features that exhibited high collinearity.

Baseline Models

1. **Model 23:** This model removed the **oldbalanceDest** and **oldbalanceOrg** features.
2. **Model 24:** This model removed the **oldbalanceDest** and **newbalanceOrig** features.
3. **Model 13:** This model removed the **newbalanceDest** and **oldbalanceOrg** features.
4. **Model 14:** This model removed the **newbalanceDest** and **newbalanceOrig** features.

Evaluation Metrics

The following evaluation metrics were used to assess the performance of the baseline models: **Accuracy, Classification Report**, **Confusion Matrix, ROC-AUC Curve.**

Conclusion

Based on the evaluation results, the M24 model, which removed the **oldbalanceDest** and **newbalanceOrig** features, was selected for further evaluation and fine-tuning. This model demonstrated superior performance compared to the other baseline models, indicating that the removal of these features did not significantly impact the model's ability to accurately predict fraudulent transactions.

3.2 Hyperparameter Tuning

This section details the hyperparameter tuning process undertaken to optimize the performance of the selected Random Forest model. Hyperparameter tuning involves adjusting specific parameters of the model to find the optimal configuration for a given dataset.

The following hyperparameters were considered for tuning:

- **n_estimators**: Number of decision trees in the forest.
- **max_depth**: Maximum depth of each tree.

Optimal Hyperparameters

After exploring various hyperparameter combinations, the following values were found to be optimal:

- **n_estimators:** 25
- **max_depth:** 5

Rationale

The decision to use **n_estimators=25** and **max_depth=5** was based on the following considerations:

- **Computational efficiency:** A smaller number of trees and a shallower maximum depth can reduce computational costs, especially for large datasets.
- **Performance trade-off:** While increasing the number of trees and depth can often improve performance, it can also lead to overfitting. The chosen values represent a balance between performance and computational efficiency.

4. Model Evaluation and Interpretation

4.1 Performance Metrics

This section presents the evaluation of the final selected model, M24, and compares its performance to the baseline models. The following metrics were used:

- **Accuracy:** Overall correct predictions.
- **Classification Score:** Precision, recall, F1-score, and support for each class.
- **Confusion Matrix:** True positives, true negatives, false positives, and false negatives.
- **ROC-AUC Curve:** Area under the receiver operating characteristic curve.

Based on the evaluation results, the M24 model, which removed the **oldbalanceDest** and **newbalanceOrig** features, was selected for further evaluation and fine-tuning. This model demonstrated superior performance compared to the other baseline models, indicating that the removal of these features did not significantly impact the model's ability to accurately predict fraudulent transactions.

4.2 Feature Importance

This section analyses the feature importance of both the **baseline model (M24**) and the **fine-tuned model (M24**), comparing their respective contributions to predicting fraudulent transactions. Feature importance provides insights into the relative significance of each feature in the models' decision-making processes.

Baseline Model

Key Findings:

- **newbalanceDest** and **oldbalanceOrg** emerged as the most important features, suggesting that account balances are crucial indicators of fraudulent activity.
- **type_TRANSFER** and **type_CASH_OUT** were also identified as significant features.
- **amount** and **step** had moderate importance.

Fine-Tuned Model

Key Findings:

- **newbalanceDest** and **oldbalanceOrg** retained their prominence, indicating their continued importance in the fine-tuned model.
- The relative importance of **type_TRANSFER** and **type_CASH_OUT** remained consistent.
- **amount** and **step** showed slight variations in importance, potentially due to the hyperparameter tuning process.

Comparison

- **Consistency:** Both models highlighted the importance of account balances (**newbalanceDest** and **oldbalanceOrg**) and transaction types (**type_TRANSFER** and **type_CASH_OUT**).
- **Feature Prioritization:** The fine-tuned model might have refined the prioritization of features based on the optimization process.
- **Impact of Tuning:** Hyperparameter tuning could have influenced the relative importance of features, especially if it affected the model's focus on specific aspects of the data.

4.3 Evaluation Metrics Results of Fine-tuned Model (M24)

The fine-tuned M24 model demonstrated exceptional performance in fraud detection, achieving an accuracy of 0.9991. This indicates that the model was highly effective in correctly classifying fraudulent and legitimate transactions.

Classification Metrics

- **Precision:** 1: The model exhibited perfect precision, meaning that when it predicted a transaction as fraudulent, it was indeed fraudulent.
- **Recall:** 1: The model achieved perfect recall, indicating that it correctly identified all fraudulent transactions.
- **F1-score:** 1: The F1-score, which is the harmonic mean of precision and recall, also reached 1, demonstrating the model's overall effectiveness.
- **Support:** 1270881: The model evaluated a total of 1,270,881 transactions.

Confusion Matrix

The confusion matrix further highlights the model's performance. It correctly classified 1,270,881 fraudulent transactions (True Positives) and 526 legitimate transactions (True Negatives). There were 1,117 false positives, indicating that the model incorrectly labelled a small number of legitimate transactions as fraudulent.

Area Under the Curve (AUC)

The AUC score of 0.9806 indicates that the model's performance was excellent in distinguishing between fraudulent and legitimate transactions. An AUC score closer to 1 signifies better performance.

Overall, the fine-tuned M24 model demonstrated exceptional performance in fraud detection, achieving high accuracy, precision, recall, F1-score, and AUC scores. These results indicate that the model is well-suited for real-world fraud detection applications.

5. Fraud Detection and Prevention Strategies

5.1 Fraudulent Customer Identification

Once the model is trained and deployed, it can be used to identify potential fraudulent customers based on their transaction patterns. By analyzing various features such as transaction type, amount, time, and account balances, the model can flag transactions that exhibit characteristics that are like known fraudulent activities.

5.2 Prevention Measures

To effectively prevent fraud, organizations can implement the following measures:

- **Transaction Monitoring:** Continuously monitor transactions in real-time to detect anomalies and suspicious activities.
- **Real-time Alerts:** Set up alerts to notify relevant personnel immediately when potential fraud is detected.
- **Manual Review:** Have a team of experts manually review flagged transactions to assess their legitimacy and take appropriate action.
- **Rule-based Systems:** Implement rule-based systems to identify and block transactions that violate predefined rules.
- **Behavioural Analytics:** Analyse customer behaviour patterns to detect deviations from normal activity and identify potential fraud.

5.3 Monitoring and Evaluation

To ensure the model's effectiveness, it is crucial to continuously monitor its performance and update it as new fraud patterns emerge. This involves:

- **Performance Metrics:** Regularly evaluate the model's accuracy, precision, recall, and F1-score using appropriate metrics.
- **False Positives and Negatives:** Analyse false positives and negatives to identify areas for improvement and refine the model.
- **Model Retraining:** Retrain the model periodically with updated data to incorporate new fraud patterns and improve its accuracy.

6. Conclusion and Future Work

Summary of Key Findings and Contributions

This project successfully developed a robust machine learning model for fraud detection using Random Forest. The model effectively identified fraudulent transactions by analysing various transaction attributes and patterns. The key findings include:

- The importance of account balances (**newbalanceDest** and **oldbalanceOrg**) and transaction types (**type_TRANSFER** and **type_CASH_OUT**) in predicting fraudulent activity.
- The effectiveness of the Random Forest algorithm in handling complex datasets and non-linear relationships.
- The ability of the model to accurately classify fraudulent and legitimate transactions.

Limitations and Areas for Future Improvement

While the model achieved satisfactory performance, there are areas for future improvement:

- **Data Quality:** The quality and completeness of the data can significantly impact the model's accuracy. Improving data quality and completeness can enhance the model's performance.
- **Model Complexity:** The Random Forest model, while effective, can be computationally expensive for large datasets. Exploring more efficient algorithms or techniques might be beneficial in certain scenarios.
- **Emerging Fraud Patterns:** Fraudsters constantly evolve their tactics. The model needs to be continuously updated to detect new fraud patterns and remain effective.

Recommendations for Implementation

To implement the fraud detection system in a real-world environment, the following recommendations are suggested:

- **Integration with Existing Systems:** Integrate the model with existing financial systems to enable real-time fraud detection.
- **User Training:** Provide training to relevant personnel on how to use the system effectively and interpret the model's outputs.
- **Regular Monitoring and Updates:** Establish a process for regular monitoring and evaluation of the model's performance, and ensure it is updated with new data and fraud patterns.
- **Risk Assessment:** Conduct a risk assessment to identify high-risk customers and transactions that require closer scrutiny.
- **Human Oversight:** Maintain human oversight to ensure that the model's decisions are not solely relied upon and that appropriate actions are taken.

By following these recommendations, organizations can effectively leverage the fraud detection system to mitigate financial risks and protect their customers from fraudulent activities.